

Docket No. AUS920030294US1

CLAIMS:

What is claimed is:

1. A method for managing the provisioning of a plurality of resources in a data processing system, said plurality of resources being a plurality of different types, said method comprising the steps of:

defining a plurality of provisioning states for each one of said plurality of different types of resources;

defining relationships among said plurality of provisioning states, said relationships describing valid transitions from ones of said plurality of states to other ones of said plurality of states; and

defining at least one task that is associated with each one of said valid transitions.

2. The method according to claim 1, further comprising the steps of:

determining that one of said plurality of resources that is one of said plurality of types should transition from a current one of said plurality of states;

determining a next one of said plurality of states utilizing said relationships;

determining at least one task that is associated with said next one of said plurality of states; and

executing said at least one task causing said one of said plurality of resources to transition to said next one of said plurality of states.

Docket No. AUS920030294US1

3. The method according to claim 1, further comprising the steps of:

generating a state diagram for each one of said plurality of different types, each one of said plurality of different types being associated with one of said state diagrams; and

each one of said state diagrams describing valid transitions for a plurality of provisioning states defined for each one of said plurality of different types.

4. The method according to claim 3, further comprising the steps of:

determining that one of said plurality of resources that is one of said plurality of types should transition from a current one of said plurality of states;

retrieving a state diagram that is associated with said one of said plurality of types; and

determining a next one of said plurality of states utilizing said retrieved state diagram.

5. The method according to claim 1, further comprising the steps of:

specifying a plurality of tasks with each one of said valid transitions;

specifying a sequence for completion for said plurality of tasks for each one of said valid transitions, said plurality of tasks being required to be completed in said sequence in order to complete each one of said valid transitions.

Docket No. AUS920030294US1

6. The method according to claim 5, further comprising:
providing said plurality of tasks in said sequence
as a module that will complete one of said valid
transitions when said module is executed; and

utilizing said module to complete said one of said
valid transitions for each one of said plurality of
different types of resources, wherein the same module is
used regardless of which resource type is being
transitioned.

7. The method according to claim 1, further comprising
the steps of:

defining said at least one task including defining a
determination of a current working condition of one of
said plurality of resources.

8. The method according to claim 1, further comprising
the steps of:

defining a plurality of task modules; and
each one of said plurality of task modules being
available for use to execute a transition from one of
said plurality of states to another one of said plurality
of states for any one of said plurality of different
types of resources.

9. The method according to claim 1, wherein said step
of defining at least one task further comprises the step
of defining a conditional task, said conditional task
requiring the completion of a second task before said
conditional task is completed.

Docket No. AUS920030294US1

10. A method in a data processing system for describing data processing system resources, said method comprising:

providing a resource hierarchy model for each type of resource, said model including a resource type level, said resource type level being divided into one or more resource implementations that are included in a resource implementation level that is below said resource type level, and a resource instance level each one or more resource implementations being divided into one or more resource instances that are included in said resource instance level that is below said resource implementation level; and

describing each particular resource utilizing a resource type, one of said one or more resource implementations included in said resource type, and one of said one or more resource instances included in said one of said one or more resource implementations.

11. A data processing system for managing the provisioning of a plurality of resources, said plurality of resources being a plurality of different types, comprising:

a plurality of provisioning states defined for each one of said plurality of different types of resources;

relationships defined among said plurality of provisioning states, said relationships describing valid transitions from ones of said plurality of states to other ones of said plurality of states; and

at least one task defined that is associated with each one of said valid transitions.

Docket No. AUS920030294US1

12. The system according to claim 11, further comprising:

said system including a CPU executing code for determining that one of said plurality of resources that is one of said plurality of types should transition from a current one of said plurality of states;

said relationships for determining a next one of said plurality of states;

said relationships for determining at least one task that is associated with said next one of said plurality of states; and

said CPU executing code for executing said at least one task causing said one of said plurality of resources to transition to said next one of said plurality of states.

13. The system according to claim 11, further comprising:

a state diagram generated for each one of said plurality of different types, each one of said plurality of different types being associated with one of said state diagrams; and

each one of said state diagrams describing valid transitions for a plurality of provisioning states defined for each one of said plurality of different types.

14. The system according to claim 13, further comprising:

Docket No. AUS920030294US1

said CPU executing code for determining that one of said plurality of resources that is one of said plurality of types should transition from a current one of said plurality of states;

a state diagram that is associated with said one of said plurality of types being retrieved; and

said retrieved state diagram for determining a next one of said plurality of states.

15. The system according to claim 11, further comprising:

a plurality of tasks being specified for each one of said valid transitions;

a sequence being specified for completion for said plurality of tasks for each one of said valid transitions, said plurality of tasks being required to be completed in said sequence in order to complete each one of said valid transitions.

16. The system according to claim 15, further comprising:

a module that includes said plurality of tasks in said sequence, said module being used to complete one of said valid transitions when said module is executed; and

said module being utilized to complete said one of said valid transitions for each one of said plurality of different types of resources, wherein the same module is used regardless of which resource type is being transitioned.

Docket No. AUS920030294US1

17. The system according to claim 11, further comprising:

said at least one task being defined including defining a determination of a current working condition of one of said plurality of resources.

18. The system according to claim 11, further comprising:

a plurality of task modules; and
each one of said plurality of task modules being available for use to execute a transition from one of said plurality of states to another one of said plurality of states for any one of said plurality of different types of resources.

19. The system according to claim 11, wherein said at least one task further comprises a conditional task, said conditional task requiring the completion of a second task before said conditional task is completed.

20. A computer program product for managing the provisioning of a plurality of resources in a data processing system, said plurality of resources being a plurality of different types, said product comprising:

instruction means for defining a plurality of provisioning states for each one of said plurality of different types of resources;

instruction means for defining relationships among said plurality of provisioning states, said relationships

Docket No. AUS920030294US1

describing valid transitions from ones of said plurality of states to other ones of said plurality of states; and

instruction means for defining at least one task that is associated with each one of said valid transitions.

21. The product according to claim 20, further comprising:

instruction means for determining that one of said plurality of resources that is one of said plurality of types should transition from a current one of said plurality of states;

instruction means for determining a next one of said plurality of states utilizing said relationships;

instruction means for determining at least one task that is associated with said next one of said plurality of states; and

instruction means for executing said at least one task causing said one of said plurality of resources to transition to said next one of said plurality of states.

22. The product according to claim 20, further comprising:

instruction means for generating a state diagram for each one of said plurality of different types, each one of said plurality of different types being associated with one of said state diagrams; and

each one of said state diagrams describing valid transitions for a plurality of provisioning states

Docket No. AUS920030294US1

defined for each one of said plurality of different types.

23. The product according to claim 22, further comprising:

instruction means for determining that one of said plurality of resources that is one of said plurality of types should transition from a current one of said plurality of states;

instruction means for retrieving a state diagram that is associated with said one of said plurality of types; and

instruction means for determining a next one of said plurality of states utilizing said retrieved state diagram.

24. The product according to claim 20, further comprising:

instruction means for specifying a plurality of tasks with each one of said valid transitions;

instruction means for specifying a sequence for completion for said plurality of tasks for each one of said valid transitions, said plurality of tasks being required to be completed in said sequence in order to complete each one of said valid transitions.

25. The product according to claim 20, further comprising:

Docket No. AUS920030294US1

instruction means for defining said at least one task including defining a determination of a current working condition of one of said plurality of resources.

26. The product according to claim 20, further comprising:

instruction means for defining a plurality of task modules; and

each one of said plurality of task modules being available for use to execute a transition from one of said plurality of states to another one of said plurality of states for any one of said plurality of different types of resources.

27. The product according to claim 20, wherein said instruction means for defining at least one task further comprises instruction means for defining a conditional task, said conditional task requiring the completion of a second task before said conditional task is completed.